

PCT/KR 03/02783

RO/KR 19.12.2003

Rec'd PCT/PTO 20 JUN 2005

REC'D 09 JAN 2004

PCT

대한민국 특허청

KOREAN INTELLECTUAL  
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 10-2002-0083165  
Application Number

출원년월일 : 2002년 12월 24일  
Date of Application DEC 24, 2002

출원인 : 한국전자통신연구원  
Applicant(s) Electronics and Telecommunications Research Ins

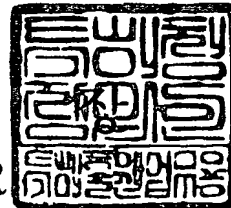
**PRIORITY DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)



2003 년 12 월 19 일

특 허 청

COMMISSIONER



BEST AVAILABLE COPY

## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0007
【제출일자】	2002.12.24
【발명의 명칭】	저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법
【발명의 영문명칭】	Doubly Journaling Store Method For Storing Data in Storage Medium
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	권태복
【대리인코드】	9-2001-000347-1
【포괄위임등록번호】	2001-057650-1
【대리인】	
【성명】	이화익
【대리인코드】	9-1998-000417-9
【포괄위임등록번호】	1999-021997-1
【발명자】	
【성명의 국문표기】	김정기
【성명의 영문표기】	KIM, Jeong Ki
【주민등록번호】	700729-1528714
【우편번호】	302-243
【주소】	대전광역시 서구 관저동 원앙마을아파트 106동 702호
【국적】	KR
【발명자】	
【성명의 국문표기】	박승민
【성명의 영문표기】	PARK, Seung Min
【주민등록번호】	580125-1162817
【우편번호】	305-755
【주소】	대전광역시 유성구 어은동 한빛아파트 102동 1302호
【국적】	KR

**【발명자】****【성명의 국문표기】**

김채규

**【성명의 영문표기】**

KIM, Chae Kyu

**【주민등록번호】**

520707-1119814

**【우편번호】**

305-345

**【주소】**

대전광역시 유성구 신성동 149-13 가람빌라 2동 201호

**【국적】**

KR

**【심사청구】**

청구

**【취지】**

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
권태복 (인) 대리인  
이화익 (인)

**【수수료】****【기본출원료】**

20 면 29,000 원

**【가산출원료】**

6 면 6,000 원

**【우선권주장료】**

0 건 0 원

**【심사청구료】**

5 항 269,000 원

**【합계】**

304,000 원

**【감면사유】**

정부출연연구기관

**【감면후 수수료】**

152,000 원

**【기술이전】****【기술양도】**

희망

**【실시권 허여】**

희망

**【기술지도】**

희망

**【첨부서류】**

1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

본 발명은 하드디스크(hard disk)나 플래시 메모리(flash memory)에 대한 저장매체에 대해 저널링(journaling) 파일 시스템을 구성할 때, 파일 시스템의 부팅 시간을 빠르게 하고 파일 시스템 회복을 쉽게 하기 위한 이중 저널링(Doubly Journaling) 저장방법에 관한 것으로, 한 종류의 데이터는 저장 매체의 처음 위치부터 저널링 방식으로 저장하고, 다른 종류의 데이터는 저장 매체의 끝 위치부터 처음 방향으로 저널링 방식으로 저장하는 것을 특징으로 한다.

**【대표도】**

도 2

**【색인어】**

이중 저널링, 저널링 파일 시스템, 플래시 메모리, garbage collection, 전반부 저널링, 후반부 저널링, 중앙점

## 【명세서】

## 【발명의 명칭】

저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법 {Doubly Journaling Store Method For Storing Data in Storage Medium}

## 【도면의 간단한 설명】

도 1은 일반적인 플래시 메모리에 파일 시스템을 구성하는 JFFS2에서 저장방법의 일 예를 보인 도면,

도 2는 본 발명에 따른 이중 저널링 저장방법에서 전반부 저널링과 후반부 저널링의 헤드가 서로 만나서 중앙점이 정해지는 실시 예를 보인 도면,

도 3은 본 발명에 따른 이중 저널링 저장방법에서 중앙점이 정해진 뒤 후반부 저널링의 헤드가 먼저 증가하여 중앙점과 만나고 중앙점을 전반부 저널링 쪽으로 이동함으로써 새로이 중앙점이 정해지는 과정의 실시 예를 보인 도면,

도 4는 본 발명의 이중 저널링 저장방법의 전반적인 수행과정을 보여주는 흐름도,

도 5는 본 발명의 이중 저널링 저장방법의 일부분으로 garbage collection(GC) 수행을 보여주는 흐름도,

도 6은 본 발명의 이중 저널링 저장방법의 GC에서 지움 블록의 개수를 결정하는 그래프.

\* 도면의 주요부분에 대한 부호의 설명 \*

H1;전반부 헤드 H2;후반부 헤드

T1;전반부 테일 T2;후반부 테일

C;중앙점

### 【발명의 상세한 설명】

### 【발명의 목적】

### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <11> 본 발명은 데이터를 저장하고 관리하고 처리하기 위한 저장 매체로 플래시 메모리(flash memory)와 하드디스크(hard disk)를 이용할 때, 이를 운영하는 파일 시스템(file system)에 관한 것이다.
- <12> 최근에 정보 산업과 이동 컴퓨팅 기술이 발전함에 따라 PDA (personal digital assistants), HPC(hand-held PC), 휴대폰, 전자책(e-book) 등이 많이 개발되고 여기에서 데이터를 저장하기 위한 목적으로 휴대가 용이하고 접근시간이 빠르며, 저전력이 요구되는 플래시 메모리를 많이 사용하고 있다.
- <13> 플래시 메모리는 일반적인 RAM(random access memory)과는 다른 특성을 가지고 있다. 플래시 메모리는 비 휘발성의 특성을 갖고 있으며, 하드디스크에 비해 견고하다. 저 전력으로 동작이 가능하며 접근 시간이 RAM과 유사할 만큼 빠르다. 또한 크기가 작아 휴대 기기에 적합하다.
- <14> 그러나, 하드디스크에 비해 가격이 5-10배 정도 비싸며, 이미 데이터가 있는 공간에 새로운 데이터를 쓰고자 할 때는 지움(cleaning) 과정을 수행한 다음에서야 다시 저장할 수 있는 단점이 있다.

- <15> 인텔(Intel co.) 회사에서 개발한 28F640J3A 모델(model) 플래시 메모리를 기준으로 할 때, 읽기 속도는 100~150 nsec(나노초)로 RAM과 유사할 정도로 빠르지만, 쓰기 속도와 지우는 속도가 상대적으로 느리다. 버퍼(buffer)를 이용한 쓰기 시간은 32bytes 버퍼를 이용할 경우 218  $\mu$  sec(마이크로초) 정도가 걸리며, 지움 블록(erase block) 단위의 쓰기 시간은 블록 당 0.8 sec 정도가 걸린다. 또한 한번에 지울 수 있는 지움 블록의 크기가 128 Kbytes로 일정하며, 상온에서 지운 다음 재 사용할 수 있는 회수가 약 10만 번으로 정해져 있다. 이렇게 한번에 지울 수 있는 플래시 메모리의 공간을 지움 블록(erase block) 또는 세그먼트(segment)라 한다.
- <16> 플래시 메모리는 셀(cell)을 구성하는 구조에 따라, NOR, NAND, AND 타입(type) 등으로 구분할 수 있으나, 주로 NOR나 NAND 플래시 메모리를 주로 사용한다. NOR 플래시는 임의 접근(random access)의 읽기 속도가 빠르고 비트 당 접근이 용이함으로 메모리 주소 공간에 직접 연결되어 CPU가 수행하는 코드(code)를 저장하는 목적으로 주로 사용되며, NAND 플래시는 상대적으로 느린 임의 접근시간 때문에 음악 파일이나 이미지 파일 등 상대적으로 큰 용량의 데이터를 한번에 저장하는 용도로 주로 이용된다. 본 발명의 방법을 설명하기 위해서 주로 OR 타입의 플래시 메모리를 기준으로 설명한다.
- <17> 이러한 플래시 메모리를 이용하여 파일 시스템을 구성한 종래 기술로는 미국 특허 번호 5,404,485의 Flash File System이란 제목의 특허가 있으며, 1995년도 USENIX 컨퍼런스(conference)의 페이지 155에서 164에 발표된 A Flash Memory Based File System란 제목의 논문이 있다.
- <18> 또한, 미국 특허 번호 6,128,630의 Journal space release for log-structured storage systems와 미국 특허 번호 6,128,630의 Data storage library array with log-structured file system which allows simultaneous write and garbage collection 등에서 발명한 로그 구조의

파일 시스템(log-structured file system)를 플래시 메모리에 적용시킨 저널링 플래시 파일 시스템(JFFS, journaling flash file system)의 종래 기술이 있다.

- <19> 로그 구조의 파일 시스템은 하드디스크에 파일 시스템을 구성하고 저장할 때, 발생한 데이터를 순차적으로 저장하는 방식이다. 이렇게 함으로써, 이전에 데이터와 새로 수정된 데이터에 대한 버전(version)을 로그 형식으로 유지할 수 있다는 것이고, 현재 문제가 발생한 데이터에 대해 이전의 데이터로 되돌아 감으로써 복구할 수 있다는 장점을 갖는다. JFFS는 로그 구조의 파일 시스템을 이용하여 플래시 메모리에 대해 파일 시스템을 구성하고 발생한 데이터를 순차적으로 저장하도록 했다.
- <20> JFFS는 미국의 Axis Communications 회사(<http://developer.axis.com/software/jffs/>)에서 개발 했으며, 버전(version) 2는 미국의 RedHat 회사(<http://sources.redhat.com/jffs2/>)에서 FSF(free software foundation)의 GPL(GNU public licenses)를 따라 개발하고 있다.
- <21> 도 1은 플래시 메모리에 파일 시스템을 구성하는 JFFS2에서 저장방법의 한 실시 예를 보인 도면이다.
- <22> 도면을 참조하면, JFFS에서 플래시 메모리에 데이터가 저장되는 한가지 예를 보이고 있다. 어떤 파일 시스템에서, 예를 들어 리눅스(Linux)의 EXT2 파일 시스템에서 디렉토리(directory) 구조가 있을 때, JFFS에 저장되는 방식은 먼저 한 디렉토리의 일반적인 특성을 담기 위해 도 1의 (a)에서처럼, 디렉토리 엔트리(Dir 1 entry)를 저장한다. 여기에 저장되는 정보는 디렉토리 노드(node)의 타입, 노드의 총길이, 헤드의 CRC(cyclic redundancy check), 부모 inode 번호, 버전 값, 노드 CRC, 이름 CRC, 디렉토리 이름 등이다. 바로 다음에 이 디렉토리에 대한 inode를 저장한다. 도 1에서는 Dir 1 inode이다. 저장되는 정보는 노드 타입, 총길이, 여러 종류의 CRC, 버전 값, 사용자 ID, 그룹 ID, 생성시간, 접근시간, 수정시간 등이다.



- <23> 위와 같은 디렉토리 엔트리와 inode는 사용자에게 보이는 정보가 아니라 파일시스템에서만 사용되는 부가 정보이다. 이를 메타 데이터(meta data)라 한다. 디렉토리 안에 있는 파일을 저장할 때도 디렉토리나 마찬가지로 파일 엔트리(File 1 entry)를 저장하고 파일 inode(File 1 inode)를 저장한다. 디렉토리나 파일은 파일 시스템에서 같은 형식으로 보는데, 디렉토리는 실제적인 데이터가 없고, 파일의 데이터(File 1 data)는 inode 다음에 저장된다.
- <24> 이런 식으로 JFFS는 플래시 메모리에 대해 디렉토리나 파일을 순차적으로(journaling) 저장하는 방식을 사용한다. 도 1의 (b)에서처럼, Dir 1 inode와 File 1 inode가 변경되어 새로운 값으로 갱신되면, 원래의 메타 데이터는 무효화(Invalid) 상태로 만들고, 새로운 데이터는 저장 위치에 순차적으로 저장한다. 무효화 상태는 실제로 데이터가 지워지는 것이 아니라, 불필요한 데이터라고 표시만 할 뿐이다. 새로운 데이터의 버전 값이 한 단계만큼 증가한다. 그러므로 새로운 데이터에 문제가 생기면, 이전 버전의 데이터가 있는 한 복구하는 것이 용이하다.
- <25> 이렇게 갱신이 되고 새로운 데이터가 저장되다 보면, 플래시의 저장 공간이 한계에 이르게 되고 저장할 공간을 확보하기 위해 플래시 메모리를 지워야 하는데, 플래시 특성 상 지움 블록(erase block) 단위로만 지워야 하기 때문에 도 1의 (c)에서처럼 유효한(valid) 데이터(Dir 1 entry)를 옮기고, 무효한 공간들로 지움 블록이 채워지면, 도 1의 (d)에서처럼 지움 과정을 수행하여 하나의 지움 블록을 지운다.
- <26> 이렇게 저장공간이 부족하여 무효한 공간을 모은 다음, 지움 블록 단위로 지워서 새로운 공간을 확보하는 과정을 Garbage Collection(GC)라 한다. 또한, 지움 과정을 수행하여 만들어진 새로운 데이터를 저장할 수 있는 공간을 자유(Free) 공간이라 한다. 이제 자유공간이 많이 확보되었으므로 계속 데이터를 저장할 수 있다.

- <27> 저장 매체로 하드디스크를 이용할 때도 비슷한 방식으로 저장이 된다. 다만, 하드디스크는 공간을 확보하기 위하여, 유효한 데이터를 이동하고 지움 블록 크기로 무효화 블록을 만드는 경우는 없다. 그러나, 하드디스크에서 파일에 대한 접근을 빠르게 하기 위해 같은 파일의 여러 조각을 모아 물리적으로 이웃한 위치에 두도록 이동하는 경우는 있다. 하드디스크는 지움 과정을 따로 수행할 필요가 없으므로 새로운 데이터를 저장할 공간이 무효화 상태이면 바로 저장하면 된다.
- <28> JFFS의 저장 방식으로 진행하다 보면, 엔트리와 inode의 위치가 파일의 데이터와 복잡하게 섞이기 때문에, 플래시 메모리를 운영체제와 연결하여 마운트(mount)하여 파일 시스템을 형성할 때, 트리(tree) 형태의 디렉토리 구조를 메모리에 만들기 위해서는 플래시 메모리의 전체 공간을 모두 읽고 메타 데이터를 찾아야 디렉토리 구조를 구성할 수 있다는 문제점이 있다.
- <29> 저장 매체를 플래시 메모리 대신 하드디스크를 이용하여 로그 구조의 파일 시스템이나 저널링 파일 시스템을 구성할 때도 같은 문제점이 발생한다. 플래시 메모리처럼 지움 블록을 맞추기 위해 데이터를 이동할 필요는 없지만, 메타 데이터들이 파일 데이터와 섞여서 전체 공간에 흩어져 있으면, 접근 속도가 플래시에 비해 40-50배 느리고, 용량이 기가 바이트(gigabyte)를 넘는 크기에서 디스크 전체를 읽어서 파일 시스템을 구성하는 것은 매우 많은 시간을 요구하게 된다.
- <30> 그리고, 하드디스크에 대해 로그 구조의 파일 시스템이나 저널링 파일 시스템을 이용하는 것은 주로 대용량의 멀티미디어 데이터를 저장하고 재생하기 위해서 사용되는데, 메타 데이터들이 섞임으로 인하여 일정 시간 당 균일한 속도의 데이터를 전송하지 못하면, 멀티미디어 용 파일 시스템으로써의 효용성이 떨어진다.

**【발명이 이루고자 하는 기술적 과제】**

- <31> 본 발명은 이러한 문제점을 해결하기 위해 메타 데이터와 일반 파일 데이터를 분리하여, 파일 데이터는 저장 매체의 처음 위치부터 저장하고, 메타 데이터는 저장 매체의 끝 위치부터 처음 방향으로 저장하는 이중의 저널링 저장방법을 제공하는데 그 목적이 있다.
- <32> 상기한 목적을 달성하기 위한 본 발명은 메타 데이터를 저장 매체의 처음 위치부터 끝 방향으로 저장하고, 파일 데이터를 끝 위치부터 처음 방향으로 저장하는 방식도 포함한다. 또한 저장 매체의 처음부터 저장되는 데이터와, 끝에서 처음 방향으로 저장되는 데이터가 메타 데이터나 파일 데이터가 아닌 다른 정보의 데이터일 수도 있다.
- <33> 본 발명에서 제시하는 이중의 저널링 형식으로 데이터를 저장하는 방법은 한 종류의 데이터는 저장 장치에 대해 처음부터 저장하고, 다른 한 종류는 끝에서 앞쪽으로 저장하는 방식이다. 처음부터 저장되는 데이터들을 전반부 저널링(front journaling)이라 하고, 끝에서 앞쪽으로 저장되는 데이터들은 후반부 저널링(rear journaling)이라 한다.
- <34> 저장이 발생하는 위치는 헤드(head)라 하며, 저널링의 뒤 부분에서 지움 과정을 수행하는 위치를 테일(tail)이라 한다. 즉, 전반부 저널링의 헤드와 테일이 존재하고 후반부 저널링의 헤드와 테일이 존재한다. 또한 전반부와 후반부의 양쪽에서 헤드가 증가하여 만나게 되는 지점을 중앙점이라 한다.
- <35> 저장할 데이터가 파일 시스템에 들어오면, 저장 매체에는 이 데이터를 물리적으로 저장할 공간과 이 데이터에 대한 파일을 형성하기 위해 필요한 파일 엔트리와 inode 등의 메타 데이터를 저장할 공간이 필요하다. 이런 경우 파일의 데이터는 전반부 저널링에 저장하고, 메타 데이터는 후반부 저널링에 저장할 수 있다. 반대의 경우로도 할 수 있다.

- <36> 이처럼 임의의 데이터를 저장 매체에 저장하고자 할 때, 본 발명의 이중의 저널링 저장 방법에서는 데이터를 어느 쪽에 저장할지를 판단하고 저장을 시작한다. 그리고, 데이터가 갱신 될 때는 이전의 데이터는 무효화시키고 새로운 데이터를 헤드의 위치에 저장한다. 데이터에 대한 삭제가 일어날 때는 데이터를 무효화 표시만 하면 된다.
- <37> 이러한 저장과 갱신과 삭제가 반복되다 보면, 저장 매체에 대해 전반부 저널링과 후반부 저널링이 만나게 되고 중앙점이 정해진다. 다시 전반부와 후반부 모두 처음 위치로 돌아가 데이터 처리를 계속하게 되면 그 다음에서는 전반부의 헤드 또는 후반부의 헤드 중에서 한쪽이 먼저 중앙점과 만나게 되는데, 이 때는 먼저 만나는 저널링에 저장할 데이터가 많다고 판단할 수 있으므로 중앙점을 상대방 쪽으로 밀어 데이터가 많은 쪽에 더 많은 저널링을 확보하도록 한다.
- <38> 플래시 메모리의 경우는 저널링 저장 방식을 이용하는 목적 중의 하나가 지움 블록에 대한 지움 회수를 고르게 안배하는 효과가 있다는 것인데, 중앙점을 이동하면 데이터가 많은 쪽의 저널링이 커짐으로써 전반부와 후반부에 대한 지움 회수를 고르게 안배할 수 있다.

#### 【발명의 구성 및 작용】

- <39> 이하, 본 발명의 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- <40> 도 2는 본 발명에 따른 이중 저널링 저장방법에서 전반부 저널링과 후반부 저널링의 헤드가 서로 만나서 중앙점이 정해지는 한 실시 예를 보인 도면이다.

- <41> 도 2a를 참조하면, 도면중 전반부 헤드는 H1으로 표시하고, 전반부 테일은 T1으로 표시하였다. 또한 후반부 저널링의 헤드는 H2, 테일을 T2라 표시하였으며, 중앙점을 C로 표시하였다.
- <42> 이러한 파일 시스템에 데이터 Data1가 들어오면, 데이터의 종류에 따라 전반부인지 후반부인지 판단하여 헤드부분에 저장하게 된다. 도2의 (a)에서 Data1은 전반부 저널링이다.
- <43> 도 2의 (b)는 데이터가 6개 저장될 때의 상태를 보이고 있다. Data1에서 Data4까지는 전반부 저널링의 데이터이고 Data5와 Data6은 후반부 저널링의 데이터이다. 그 다음 수행으로, Data2와 Data6이 지워졌다. 저널링 저장 방식의 특성 상 데이터가 실제로 지워진 것이 아니라, 무효화(invalid) 상태로 표시된 것 뿐이다.
- <44> 도 2의 (c)에서는 저장 매체에 대해 저장공간이 부족하여 Garbage collection(GC)이 발생한 상태를 보이고 있다. Data1은 유효한(valid) 데이터이므로 헤드부분으로 옮긴 다음 이전의 Data1을 무효화시키고, 이전에 무효화된 Data2와 함께 지움 블록의 크기를 넘어서면, 실제로 지움 과정을 수행함으로써 자유 공간을 확보한다.
- <45> 이렇게 실제로 지우는 과정을 수행하는 것은 플래시 메모리에서 수행할 때이고, 디스크를 사용할 때는 지움 과정이 따로 필요하지 않다. 이렇게 자유공간이 확보되면, 전반부 저널링의 끝부분인 테일(T1)은 Data3의 위치로 이동된다. 마찬가지로 Data6도 무효화되고 지움 블록 크기가 되어 지워지면, 후반부의 저널링의 테일 (T2)도 이동된다.
- <46> 이 상태에서 새로운 데이터 Data7을 저장하고자 한다. 이 데이터는 전반부 저널링의 데이터로 전반부 헤드(H1)에 저장하고자 하지만, 공간이 부족하다. 그러므로, 도 2의 (d)에서 처

럼, 저장할 수 있는 만큼인 D7-1만 저장하고 전반부 헤드 (H1)이 처음으로 돌아와서 나머지 D7-2를 저장한다.

- <47> 이런 방식으로 전반부 저널링이 후반부 저널링과 만나면, 원형으로 처음 위치로 돌아와서 저장을 하게 된다. 또한 전반부의 헤드(H1)와 후반부의 헤드(H2)가 만났던 지점이 중앙점(C)으로 설정된다. 후반부 헤드(H2)도 끝 위치로 돌아가서 저장을 기다린다. 이런 방식으로 전반부 저널링과 후반부 저널링에 데이터가 저장되며, 중앙점(C)이 정해진다.
- <48> 이런 방법으로 계속 데이터를 저장하다 보면, 전반부 헤드(H1)나 후반부 헤드(H2) 중에서 하나가 먼저 다시 중앙점(C)의 위치에 도달하게 되는데, 이 경우 먼저 도달하는 저널링 부분이 저장할 데이터가 많다고 판단할 수 있다. 그러므로 중앙점을 상대방 쪽으로 옮겨 먼저 중앙점(C)에 도착하는 저널링의 공간을 증가시켜주도록 한다.
- <49> 도 3은 이중 저널링 저장방법에서 중앙점이 정해진 뒤 후반부 저널링의 헤드가 먼저 증가하여 중앙점과 만나고 중앙점을 전반부 저널링 쪽으로 이동함으로써 새로이 중앙점이 정해지는 과정의 하나의 실시 예를 보인 도면이다.
- <50> 도면을 참조하면, 도 3의 (a)에서 Data3과 Data5가 무효화되고, 후반부 저널링의 데이터 저장이 활발하여 Data8과 Data9가 저장되고, Data10을 저장하려고 하지만 공간이 부족하다. 후반부에서 먼저 중앙점(C)에 도달한 것이다. 후반부의 헤드(H2)가 전반부의 헤드(H1)를 만나지 않았으므로, 전반부에 저장할 공간이 충분히 있는 한은 중앙점(C)을 전반부 쪽으로 이동시킨다.
- <51> 전반부에 데이터가 충분히 있다는 판단은 도 5의 GC 수행에서 하게 된다. 이렇게 하기 위해 도 5의 (b)에서처럼 전반부의 데이터인 D7-2를 전반부 헤드(H1)로 이동시키고, 중앙점(C)

도 이동시킨다. 이 때, 전반부에서 이동하는 데이터의 단위는 플래시 메모리의 경우 지움 블록 단위이다. 왜냐하면, 이동한 뒤 지움 과정을 수행하여 자유 공간을 만든 다음에 새로운 데이터를 저장할 수 있기 때문이다. 자유 공간이 만들어지면, 새로운 데이터 Data10을 저장한다. 이러한 중앙점(C)의 이동은 전반부의 헤드(H1)를 만날 때까지 또는 전반부의 여유 데이터 저장 공간이 충분히 있을 때까지 이동된다.

<52> 도 4는 본 발명에 따른 이중 저널링 저장방법의 전반적인 수행과정을 보인 흐름도이다.

<53> 도면을 참조하면, 본 발명은 데이터가 저장되는 과정과 중앙점이 결정되는 과정, 중앙점이 이동되는 과정을 흐름도를 통해 보이고 있다.

<54> 먼저 기본적으로 전반부의 헤드(H1), 후반부의 헤드(H2), 전반부의 테일(T1), 후반부의 테일(T2) 및 중앙점(C)의 초기값은 0이다.(S101)

<55> 파일 시스템에 대한 데이터 저장 요구는 버퍼를 통해 이루어지는데, 이러한 상태에서 저장 매체에 저장 요구(S102)가 이루어지면, 도 5의 GC를 통해 저장 공간이 충분한지 확인한다.(S103)

<56> 만약, GC에 의해 저장 매체의 공간을 확인한 결과 상기 저장 매체의 저장 공간이 충분하면 도 4의 흐름도로 바로 돌아와 수행을 계속하고, 충분하지 않으면 저장공간을 확보하고 난 다음 도 4의 흐름도로 바로 돌아와 수행을 계속한다.

<57> 이러한 상태에서 중앙점(C)이 0인지 확인(S104)하게 되는데, 이것은 중앙점이 처음 결정되는지를 확인하기 위해서이다. 중앙점(C)이 처음 결정되는 경우는 헤드가 상대방 헤드와 만나는 것을 확인해야 하지만, 이미 중앙점이 형성된 경우는 헤드가 중앙점을 먼저 만나는지 확인해야 하기 때문이다.

- <58> 본 발명은 단계 104의 판단 결과 중앙점(C)이 0일 경우 또는 0이 아닐 경우 각각의 경우에 대해 데이터가 전반부 저널링 데이터인지 확인(S105)하고, 전반부인 경우 데이터(H1+S)를 저장한다. 이 상태에서 저장 데이터(H1+S)가 후반부 헤드(H2)와 만나게 되는지를 확인(S106)하고, 반대로 데이터(H1+S)가 전반부 저널링 데이터가 아닐 경우 후반부는 전반부와 대칭적으로 수행한다.(S107)
- <59> 만약, 데이터(H1+S)가 후반부 헤드(H2)를 넘지 않으면 그대로 저장(S108)하면 되고, 넘는 경우 데이터 크기(S) 중 가능한 크기를 저장하고 전반부 헤드(H1)를 처음 위치로 두고 나머지를 저장하고, 만나는 곳이 중앙점(C)으로 정해진다. 이처럼 데이터가 저장되면 중앙점(C)이 정해지게 되고, 따라서 중앙점(C) 값은 더 이상 0이 아니다.
- <60> 또한, 단계 104에서 중앙점(C)이 0이 아닐 경우에도 0인 경우와 마찬가지로 데이터가 전반부 저널링 데이터인지 확인(S110)한다. 만약 전반부일 경우에는 데이터(H1+S)를 저장한다. 이 상태에서 저장 데이터(H1+S)가 중앙점(C)과 만나는지 확인(S111)하고, 반대 데이터(H1+S)가 전반부 저널링 데이터가 아닐 경우 단계 S106과 같이 후반부는 전반부와 대칭적으로 수행한다.(S113)
- <61> 단계 111에서 데이터(H1+S)가 중앙점(C)을 넘지 않으면 전반부 헤드(H1)에 데이터를 저장하고 공간을 확보한다.(S112)
- <62> 반대로, 단계 111에서 데이터(H1+S)가 중앙점(C)을 넘을 경우 저장할 수 있는 만큼만 전반부 헤드(H1)와 중앙점(C)사이에 데이터를 저장(S114)하고, 이후 중앙점(C)을 후반부 저널링 쪽으로 옮긴 다음 저장한다.



- <63> 이러한 상태에서 후반부에 공간이 충분한지 확인(S115)한다. 만약 후반부에 공간이 충분할 경우 데이터를 후반부 헤드(H2)로 옮기고 공간을 지우고(S116), 지운 공간에 데이터를 저장한다.(S117)
- <64> 한편, 단계 115에서 후반부 저널링에 더 이상 충분한 공간이 없을 경우 후반부의 가능한 공간을 지우고, 중앙점(C)를 그 만큼 이동시킨 후 데이터를 저장하고 저장하지 못한 나머지 데이터는 전반부 헤드(H1)를 처음위치로 두고 저장(S118)하고, 이러한 저장 과정이 끝나면, 다시 도 5에서 처럼 GC 수행과정을 부른다.
- <65> 도 5는 이중 저널링 저장방법의 일부분으로 가비지 컬렉션(GC; garbage collection) 수행 과정을 보인 흐름도이다.
- <66> 도 5의 GC 수행 과정에서는 일단, 데이터 저장에 대한 요구가 들어오면, GC수행되어 데이터를 저장할 만큼 자유공간이 있는지 확인한다. 저장할 데이터는 일단 저장 버퍼(buffer)에 존재한다. 저장할 데이터 크기의 자유공간이 없으면, 저널링 방법에 의해 데이터를 옮기고 무효화 공간을 지움 블록만큼 모아 지움 과정을 수행한 다음, 새로운 데이터에 대한 최소한의 공간이 확보되면, 먼저 저장을 수행한다. 여기까지가 도 4에서 처음 부분 GC를 부른 경우이다.(도 4의 S101-S103)
- <67> 그리고, 일정한 데이터 크기에 대해 지움 블록 프리 블록이 존재한 상태(S201)에서 버퍼에 데이터가 있는지 확인(S202)한 후, 만약 버퍼에 데이터가 있을 경우 그 데이터가 전반부 저널링인가 확인(S203)한다. 단계 203에서 데이터가 전반부 저널링이 아닐 경우 후반부도 전반부와 같은 방식으로 수행(S204)하고, 반대로 데이터가 전반부 저널링일 경우 자유공간이 충분한지 확인(S205)한다.

- <68> 단계 205에서 자유공간이 충분하면 GC 수행을 종료하고, 반대로 자유공간이 충분하지 않으면 전반부 테일(T1)에서 유효한 공간을 전반부 헤드(H1)으로 옮기고 지움 블록에 무효화 공간을 확보하고 지운다.(S206)
- <69> 한편, 삽입 데이터에 대한 저장의 수행된 뒤 저장 공간 내에 충분한 공간을 확보하기 위해 다시 한번 GC가 불러지는데, 이 때는 도 6의 그래프에 의해 지워야 할 지움 블록의 개수를 결정한다. 이 때는 저장 버퍼에 저장할 데이터가 없을 경우이다.
- <70> 이러한 상태에서 도 6과 같이 결정 그래프에서  $N_2$  이상의 자유(free) 지움 블록을 확보하고 있으면, 더 이상 지움 과정이 발생하지 않는다.  $N_2$ 는 보통 전체 저장 공간에 대해 약 10% 정도로 설정한다.
- <71> 자유 지움 블록의 개수가  $N_1$ 에서  $N_2$  사이에 있을 때는 최대한  $N_2$  이상의 자유 지움 블록을 확보하기 위하여 그래프의 지움 지움 블록 개수만큼 지운다.(S208)
- <72> 단계 207에서 자유 지움 블록의 개수가  $N_1$  이하이면, 어떤 연산보다 우선하여 지움 블록을 확보한다. 지움 연산을 수행한 뒤로도 자유 지움 블록의 개수가  $N_1$  이하이면 더 이상의 저장공간이 없다는 뜻이다.(S209-S211)
- <73> 처음 GC에서부터 충분한 자유 공간을 확보하지 않는 이유는 지움 과정이 읽거나 쓰기에 비해 시간을 많이 소모하는데, 데이터 저장에 대한 요구가 들어 왔을 때, 자유 공간을 충분히 확보하기 위해서는 데이터 저장에 대한 수행 시간(latency time)이 길어지기 때문이다.

- <74> 또한, 대부분의 플래시 메모리는 지움 과정을 수행하다가 읽기 및 쓰기 연산의 요구가 들어오면, 지움 과정을 보류하고(suspend) 우선 순위가 높은 연산을 수행하도록 하는 기능이 있기 때문에 이를 활용할 수 있다.

### 【발명의 효과】

- <75> 이상에서 설명한 바와 같이 본 발명의 실시예에서는 플래시 메모리나 하드디스크에 데이터를 저장할 때, 이중의 저널링 방식으로 저장하는 방법을 기술하고 있다.
- <76> 실시예에서와 같이 이중 저널링 저장방법을 이용할 경우, 종류가 같은 데이터를 일정 영역에 유지하기 때문에 빠른 데이터 접근 시간을 이룰 수 있으며, 플래시 메모리에서 지움 회수를 플래시 메모리 공간에 고르게 안배할 수 있다.
- <77> 또한 저널링 저장 방식의 특징에 따라 전원 오류 등에 의해 데이터 오류가 발생한 경우, 이전 버전으로의 데이터 복구가 용이함으로 데이터에 대한 신뢰성을 보장하는 효과가 있다.
- <78> 이상에서 설명한 것은 본 발명에 따른 저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법을 설명한 하나의 실시 예에 불과한 것으로서, 본 발명은 상기한 실시 예에 한정되지 않고, 이하의 특허청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변경 실시가 가능한 범위까지 본 발명의 기술적 사상이 미친다고 할 것이다.

**【특허청구범위】****【청구항 1】**

저장 매체에 데이터를 저장하기 위한 방법에 있어서,

상기 저장 매체의 저장공간 처음 위치와 끝 위치로부터 각각 중앙쪽으로 데이터를 저장해 가는 것을 특징으로 하는 저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법.

**【청구항 2】**

제 1항에 있어서,

상기 저장 매체에 종류나 성격이 서로 다른 종류의 데이터가 저장될 때, 전반부 저널링과 후반부 저널링으로 각각 구분되어 저장되는 것을 특징으로 하는 저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법.

**【청구항 3】**

제 2항에 있어서,

데이터가 저장되어 전반부 저널링과 후반부 저널링이 중앙점에서 만날 경우 처음 위치에서 다시 데이터가 저장되는 것을 특징으로 하는 저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법.

**【청구항 4】**

제 2항에 있어서,

상기 전반부 저널링의 헤드와 상기 후반부 저널링의 헤드가 만나 중앙점이 처음으로 형성되고, 이후, 중앙점이 두 번째 이상 형성되는 경우에 대해 전반부 또는 후반부 헤드가 중앙점을 다시 만날 경우, 중앙점이 상대방 저널링 쪽으로 이동되는 것을 특징으로 하는 저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법.

#### 【청구항 5】

저장 매체에 데이터 삽입시 저널링 저장방법에 있어서,

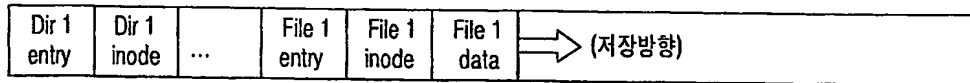
상기 저장매체에 데이터 삽입에 대한 요구가 들어왔을 때 GC를 수행하여 저장공간이 충분하지 않을 경우 데이터를 이동하여 지움 과정을 수행하는 단계; 및

데이터 저장이 완료된 후 상기 저장 매체에 충분한 공간이 있는지 확인하여 다음 삽입 데이터를 저장하기 위한 공간을 확보하는 단계;를 포함하는 것을 특징으로 하는 저장 매체에 데이터를 저장하기 위한 이중 저널링 저장방법.

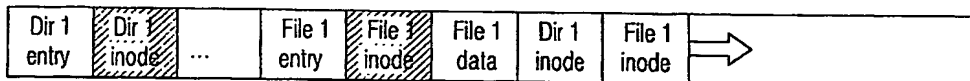
## 【도면】

## 【도 1】

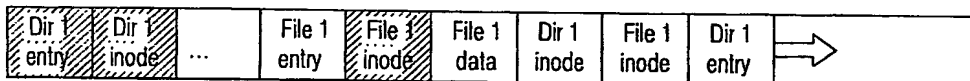
&lt;플래시메모리&gt;



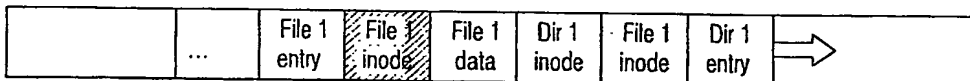
(a) 초기 순차적 저장상태



(b) "Dir 1 inode"와 "File 1 inode"가 변경된 상태

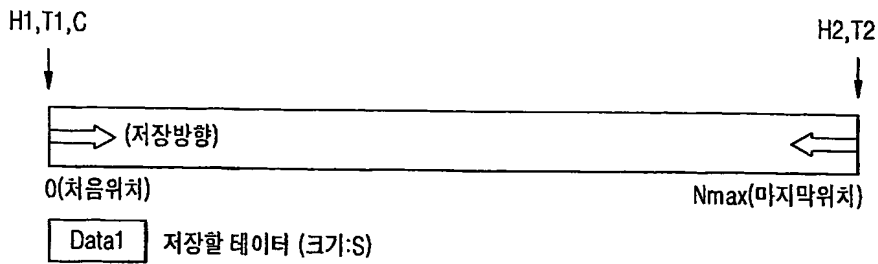


(c) Garbage Collection를 위해 "Dir 1 entry"를 이동한 상태

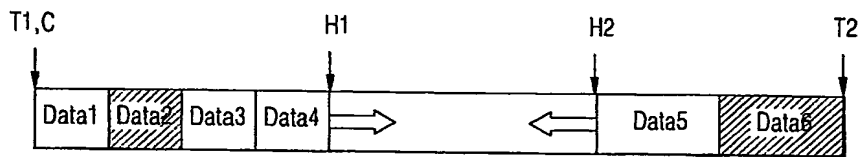


(d) GC 이후 한 지움 블록을 지운 상태

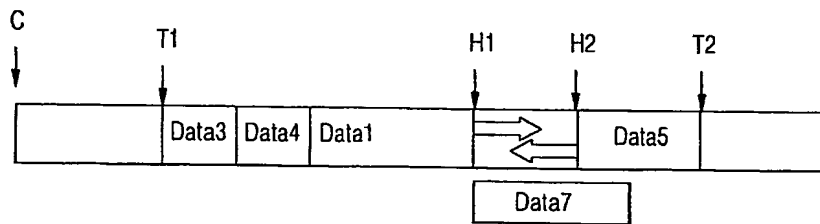
【도 2】



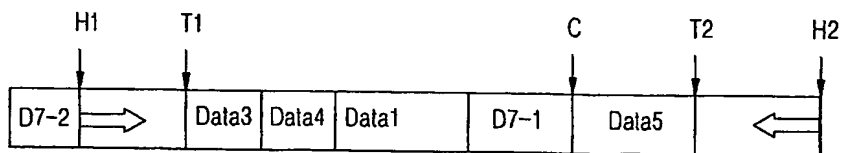
(a) 크기 S인 데이터를 저장하고자 하는 상태



(b) 데이터 6개를 저장하고 "Data2"와 "Data6"이 지워진 상태

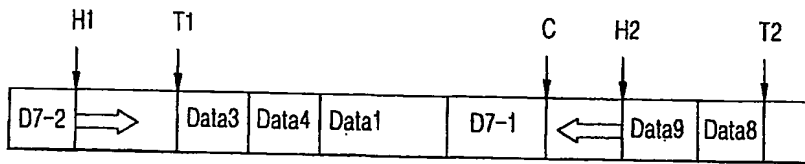


(c) GC를 위해 "Data1"를 옮기고, 무효공간을 지운다음 "Data7"를 저장하고자 하는 상태

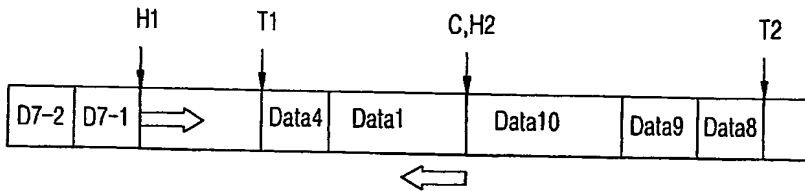


(d) "Data7"이 저장되고 중앙점이 정해진 상태

【도 3】



(a) "Data10"를 저장하고자 하지만 공간이 없는 상태



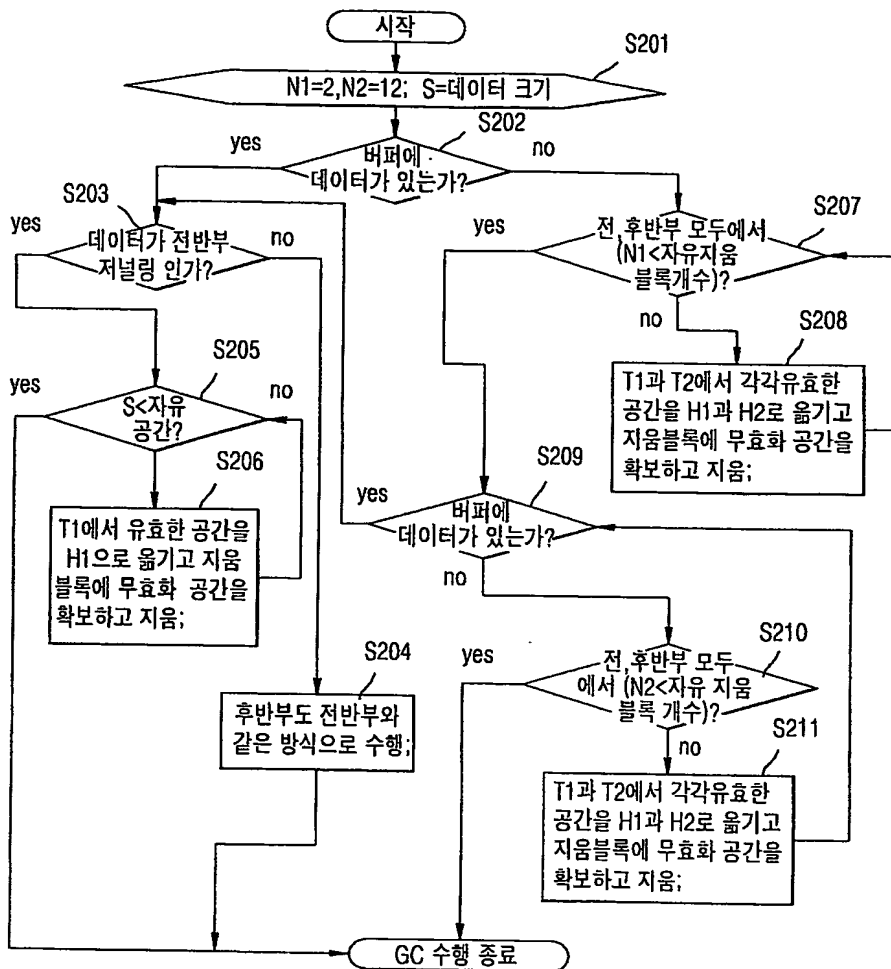
(b) 후반부 저널링의 헤드가 중앙점을 이동시킨 상태



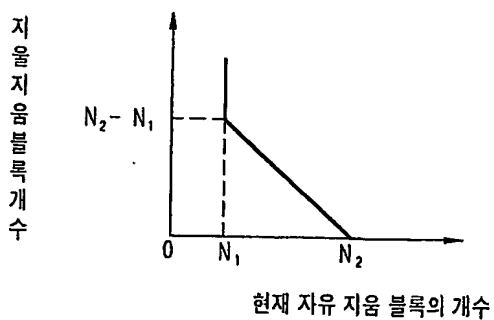
```

graph TD
    Start([시작]) --> S101[/H1=0, T1=0; H2=0, T2=0; C=0/]
    S101 --> S102[버퍼(buffer)에 데이터(크기:S) 두고 저장요구]
    S102 --> S103[GC수행(저장공간 충분한지 확인)]
    S103 --> S104{C=0 인가?}
    S104 -- yes --> S105{데이터가 전반부  
저널링인가?}
    S104 -- no --> S110{데이터가 전반부  
저널링인가?}
    S105 -- yes --> S106{H1+S ≤ H2?}
    S105 -- no --> S107[후반부는 전반부와  
대칭적으로 수행;]
    S106 -- yes --> S108[H1에 데이터 저장;  
H1=H1+S;]
    S106 -- no --> S109[H1에 (H2-H1)만큼  
데이터 저장;  
C=H2;  
H1=0;  
H1에 S-(H2-H1)만큼  
데이터 저장;  
H1=H1+S-(H2-H1);]
    S107 --> S108
    S108 --> S117[지운공간에  
S-(C-H1)크기의 삽입  
데이터 저장;]
    S109 --> S117
    S110 -- yes --> S111{H1+S ≤ C?}
    S110 -- no --> S113[후반부는 전반부와  
대칭적으로 수행;]
    S111 -- yes --> S112[H1에 데이터 저장;  
H1=H1+S;]
    S111 -- no --> S114[H1에 (C-H1)만큼  
데이터 저장;  
H1=C;]
    S112 --> S116[후반부에서 C에서부터  
S-(C-H1)만큼  
데이터를 H2로 옮기고  
공간 지움;  
C=C+S-(C-H1);]
    S114 --> S115{(C+S-(C-H1)) < H2  
and 후반부에 공간  
충분한가?}
    S115 -- yes --> S116
    S115 -- no --> S118[후반부의 가능한 공간  
만큼 지우고, C를 그  
만큼 이동시킴;  
그 만큼의 데이터 저장;  
H1=0;  
데이터의 나머지 저장;  
H1=나머지 크기;]
    S116 --> S117
    S117 --> S119[GC 수행(충분한 공간 확보)]
    S118 --> S119
    S119 --> End([한 데이터 저장 종료])
  
```

【도 5】



【도 6】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**